G 6

```
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR        AAAAAAAAA     RRRRRRRRRRRR
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR        AAAAAAAAA     RRRRRRRRRRRR
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR        AAAAAAAAA     RRRRRRRRRRRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR     AAA         AAA  RRRRRRRRRRRR
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR     AAA         AAA  RRRRRRRRRRRR
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR     AAA         AAA  RRRRRRRRRRRR
LLL                    III        BBB         BBB  RRR   RRR        AAAAAAAAAAAAAAA  RRR   RRR
LLL                    III        BBB         BBB  RRR   RRR        AAAAAAAAAAAAAAA  RRR   RRR
LLL                    III        BBB         BBB  RRR   RRR        AAAAAAAAAAAAAAA  RRR   RRR
LLL                    III        BBB         BBB  RRR      RRR     AAA         AAA  RRR      RRR
LLL                    III        BBB         BBB  RRR      RRR     AAA         AAA  RRR      RRR
LLL                    III        BBB         BBB  RRR      RRR     AAA         AAA  RRR      RRR
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR  AAA         AAA  RRR         RRR
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR  AAA         AAA  RRR         RRR
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR  AAA         AAA  RRR         RRR
```

```
_$2

Val
---
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
```

```
**FILE**ID**COMPRESS                                          M  8









    CCCCCCCC      OOOOOO      MM      MM  PPPPPPPP    RRRRRRRR    EEEEEEEEEE    SSSSSSSS    SSSSSSSS
    CCCCCCCC      OOOOOO      MM      MM  PPPPPPPP    RRRRRRRR    EEEEEEEEEE    SSSSSSSS    SSSSSSSS
  CC            OO      OO    MMMM  MMMM  PP      PP  RR      RR  EE          SS          SS
  CC            OO      OO    MMMM  MMMM  PP      PP  RR      RR  EE          SS          SS
  CC            OO      OO    MM  MM  MM  PP      PP  RR      RR  EE          SS          SS
  CC            OO      OO    MM  MM  MM  PP      PP  RR      RR  EE          SS          SS
  CC            OO      OO    MM      MM  PPPPPPPP    RRRRRRRR    EEEEEEEE      SSSSSS      SSSSSS
  CC            OO      OO    MM      MM  PPPPPPPP    RRRRRRRR    EEEEEEEE      SSSSSS      SSSSSS
  CC            OO      OO    MM      MM  PP          RR  RR      EE                  SS          SS
  CC            OO      OO    MM      MM  PP          RR  RR      EE                  SS          SS
  CC            OO      OO    MM      MM  PP          RR    RR    EE                  SS          SS    ::::
  CC            OO      OO    MM      MM  PP          RR    RR    EE                  SS          SS    ::::
    CCCCCCCC      OOOOOO      MM      MM  PP          RR      RR  EEEEEEEEEE  SSSSSSSS    SSSSSSSS      ::::
    CCCCCCCC      OOOOOO      MM      MM  PP          RR      RR  EEEEEEEEEE  SSSSSSSS    SSSSSSSS      ::::



  LL              IIIIII        SSSSSSSS
  LL              IIIIII        SSSSSSSS
  LL                II        SS
  LL                II        SS
  LL                II        SS
  LL                II        SS
  LL                II          SSSSSS
  LL                II          SSSSSS
  LL                II                SS
  LL                II                SS
  LL                II                SS
  LL                II                SS
  LLLLLLLLLL      IIIIII      SSSSSSSS
  LLLLLLLLLL      IIIIII      SSSSSSSS
```

```
0 MODULE lib_compress (                          ! Compress the library
0                 LANGUAGE (BLISS32),
0                 IDENT = 'V04-000'
0                 ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 !
1 ! FACILITY:  Library command processor
1 !
1 ! ABSTRACT:
1 !
1 !       The VAX/VMS librarian is invoked by DCL to process the LIBRARY
1 !       command.  It utilizes the librarian procedure set to perform
1 !       the actual modifications to the library.
1 !
1 ! ENVIRONMENT:
1 !
1 !       VAX native, user mode.
1 !
1 !--
1 !
1 !
1 ! AUTHOR:  Benn Schreiber,       CREATION DATE:  22-June-1979
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-003 MCN0145         Maria del C. Nasr       08-Feb-1984
1 !               When using the /COMPRESS qualifier, if the input library
1 !               is in data reduced format, do not expand the new one.
1 !               Expansion will only be done when /DATA=EXPAND is used.
1 !
1 !       V03-002 GJA0064         Greg Awdziewicz         26-Jan-1984
```

```
1 !                     Allow benign compression of an empty library.
1 !
1 !             V03-001 JWT0056         Jim Teague              16-Sep-1982
1 !                     Equipped with DCX interface for /COMPRESS=REDUCE.
1 !
1 !             V02-007         RPG0037         Bob Grosso              15-Jan-1982
1 !                     Use library history attributes rather than default.
1 !
1 !             V02-006         RPG0036         Bob Grosso              18-Dec-1981
1 !                     Improve error reporting with update history
1 !
1 !             V02-005         RPG0035         Bob Grosso              7-Aug-1981
1 !                     lib$gl_ctlmsk now a quadword.
1 !
1 !             V02-004         RPG0034         Bob Grosso              30-Jul-1981
1 !                     Support CREATE=KEEP.
1 !
1 !             V02-003 BLS0029         Benn Schreiber          23-Dec-1980
1 !                     Change messages to use message compiler.
1 !
1 !             V02-002 RPG0004         Bob Grosso              3-Sep-1980
1 !                     Exit read or write loops and print end of module header
1 !                     and continue compressing.
1 !--
1
1
```

```
;     85        0084  1 LIBRARY
;     86        0085  1         'SYS$LIBRARY:STARLET.L32';
;     87        0086  1 REQUIRE
;     88        0087  1         'PREFIX';
;     89        0271  1 REQUIRE
;     90        0272  1         'LIBDEF';
;     91        0560  1 REQUIRE
;     92        0561  1         'LBRDEF';
;     93        1152  1
;     94        1153  1 EXTERNAL ROUTINE
;     95        1154  1     lib_get_mem,
;     96        1155  1     lbr$dcx_map: addressing_mode (general),
;     97        1156  1     lbr$get_history : ADDRESSING_MODE (GENERAL),!Get the library history
;     98        1157  1     lbr$put_history : ADDRESSING_MODE (GENERAL),!Replace history
;     99        1158  1     lbr$get_index : ADDRESSING_MODE (GENERAL),  !Call routine for index entries
;    100        1159  1     lbr$find : ADDRESSING_MODE (GENERAL),       !Find module by RFA
;    101        1160  1     lbr$lookup_key : ADDRESSING_MODE (GENERAL), !Lookup key in index
;    102        1161  1     lbr$insert_key : ADDRESSING_MODE (GENERAL), !Insert new key into index
;    103        1162  1     lbr$put_end : ADDRESSING_MODE (GENERAL),    !Terminate writing module text
;    104        1163  1     lbr$set_index : ADDRESSING_MODE (GENERAL),  !Set index number to use
;    105        1164  1     lbr$set_module : ADDRESSING_MODE (GENERAL), !Read/update module header
;    106        1165  1     lbr$get_record : ADDRESSING_MODE (GENERAL), !Read text record
;    107        1166  1     lbr$put_record : ADDRESSING_MODE (GENERAL), !Write text record
;    108        1167  1     lbr$search : ADDRESSING_MODE (GENERAL),     !Search an index for an RFA
;    109        1168  1     lbr$open : ADDRESSING_MODE (GENERAL),       !Open library
;    110        1169  1     lbr$close : ADDRESSING_MODE (GENERAL),      !Close library
;    111        1170  1     lbr$ini_control : ADDRESSING_MODE (GENERAL),!Initialize control index
;    112        1171  1     lbr$insert_time : ADDRESSING_MODE (GENERAL),!Set module insert date/time
;    113        1172  1     lib_log_op,                                 !Log operation
;    114        1173  1     lib_create_lib;                             !Create output library
;    115        1174  1
;    116        1175  1 EXTERNAL
;    117        1176  1     lbr$gl_control : REF BBLOCK ADDRESSING_MODE (GENERAL), !Librarian control table address
;    118        1177  1     lbr$gl_rmsstv : ADDRESSING_MODE (GENERAL),  !RMS STV from librarian
;    119        1178  1     lib$gl_type,                                !Type of library
;    120        1179  1     lib$al_hdrlen : VECTOR [,LONG],             !Lengths of various module headers
;    121        1180  1     lib$al_ascbinf : VECTOR [,LONG],            !Key lengths
;    122        1181  1     lib$gl_keysize,                             !Max size of key in library
;    123        1182  1     lib$gl_libctl : BLOCK [2],                  !Input library control index
;    124        1183  1     lib$gl_libfdb : REF BBLOCK,                 !Pointer to library FDB
;    125        1184  1     lib$gl_outfdb : REF BBLOCK,                 !Pointer to output library FDB
;    126        1185  1     lib$gl_ctlmsk : BLOCK [1],                  !Librarian control flags
;    127        1186  1     lib$gl_cre8flags : BITVECTOR,               !Compress option flags
;    128        1187  1     lib$gl_allgbls,                             !Number of globals to allocate in new library
;    129        1188  1     lib$gl_allmods,                             !Number of modules to allocate in new library
;    130        1189  1     lib$gl_allksz,                              !Size of keys in new library
;    131        1190  1     lib$gl_allhis,                              !Max number of history records in new library
;    132        1191  1     lib$gl_objgsdix,                            !Index number of object globals
;    133        1192  1     lib$gl_modnamix;                            !Index number of module names
;    134        1193  1
;    135        1194  1 EXTERNAL LITERAL
;    136        1195  1     lbr$_nulidx,                                ! Index is empty.
;    137        1196  1     lib$_emptylibrary,                          ! An empty library is to be
;    138        1197  1                                                 ! compressed.
;    139        1198  1     lib$_cnvrting,                              !Converting info message
;    140        1199  1     lib$_histerr,                               !Error in update history
;    141        1200  1     lib$_indexerr,                              !Some strange index error
;  142         1201  1         lib$_initerr,                                     !Initialization error
;  143         1202  1         lib$_inserted,                                    !Module inserted
;  144         1203  1         lib$_inserterr,                                   !Error inserting into index
;  145         1204  1         lib$_lookuperr,                                   !Error looking up module
;  146         1205  1         lib$_mhderr;                                      !Module header error
;  147         1206  1
;  148         1207  1    FORWARD ROUTINE
;  149         1208  1         lib_put_history,                                  !Copy over the history records.
;  150         1209  1         get_index_if_not_empty,                           ! Call Lbr$Get_index.
;  151         1210  1         copymodule,                                       !Copy one object module
;  152         1211  1         enterglobals;                                     !Enter globals for obj lib
;  153         1212  1
;  154         1213  1    GLOBAL
;  155         1214  1         dcx_map_desc : VECTOR [2];
;  156         1215  1
;  157         1216  1    OWN
;  158         1217  1         curindex,                                         !Current index being searched
;  159         1218  1         newtxtrfa : BBLOCK [dsc$c_s_bln],                 !Module RFA in new library
;  160         1219  1         outlibindex,                                      !Control index for output library
;  161         1220  1         func : LONG INITIAL  (lbr$c_create);              !Function to create library
```

```
LIB_
V04-

; Ro

;
;
;
;
;
;
;
;
;

; R
```

```
1 GLOBAL ROUTINE lib_comprs_lib  (after_func) =
2 BEGIN
2
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       This routine compresses one library into another.
2 !
2 ! CALLING SEQUENCE:
2 !
2 !       status = lib_comprs_lib (after_func)
2 !
2 ! INPUT PARAMETERS:
2 !
2 !       after_func      is the function (lbr$c_read, lbr$c_update) to open the compressed library with
2 !                       after the compress has been completed
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       lib$gl_libfdb   is the pointer to the library (input FDB)
2 !       lib$gl_outfdb   is the pointer to the output FDB
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       lib$gl_libfd is changed to point to the output FDB
2 !
2 ! SIDE EFFECTS:
2 !       NONE
2 !
2 !--
2
2 LOCAL
2     usrmodhdrlen,                             ! temp store expansion size of module header
2     header : REF BBLOCK,
2     status;
2
2 BIND
2     libdesc = lib$gl_libfdb [fdb$l_namdesc] : BBLOCK,            !Name the filename descriptor
2     outdesc = lib$gl_outfdb [fdb$l_namdesc] : BBLOCK,            ! for input and output libraries
2     libnamblk = lib$gl_libfdb [fdb$t_nam] : BBLOCK,              !Name the NAM blocks
2     outnamblk = lib$gl_outfdb [fdb$t_nam] : BBLOCK;              ! ...
2
2 !
2 ! Determine what create options we need to derive from input library
2 ! and do it.
2 !
2 header = .lbr$gl_control [lbr$l_hdrptr];                         !point to library header
2 IF NOT .lib$gl_cre8flags [lib$c_opt_gbls]                         !Globals specified by option?
2 THEN lib$gl_allgbls = .header [lhd$l_idxcnt] - .header [lhd$l_modcnt]; !No--compute from header
2 IF NOT .lib$gl_cre8flags [lib$c_opt_mods]                         !Modules specified by option
2 THEN lib$gl_allmods = .header [lhd$l_modcnt] + .header [lhd$l_idxovh];
2 IF NOT .lib$gl_cre8flags [lib$c_opt_ksz]                          !Key size specified?
2 THEN IF .lib$gl_ctlmsk [lib$v_oldlib]
2     THEN lib$gl_allksz = .lib$al_ascbinf [.lib$gl_type]          ! if old library, then get new size
3     ELSE BEGIN
3 !
```

```
;  220         1278  3 ! Get size of keys from input library if new format
;  221         1279  3 !
;  222         1280  3         BIND
;  223         1281  3             indexdesc = .header + lhd$c_idxdesc : BBLOCK;          !Point to first index descriptor
;  224         1282  3
;  225         1283  3         lib$gl_allksz = .indexdesc [idd$w_keylen] - 1;          !Get size of keys minus count byte
;  226         1284  2         END;
;  227         1285  2 lib$gl_keysize = .lib$gl_allksz;                                !Set key size for future reference
;  228         1286  2 lib$gl_cre8flags [lib$c_opt_ksz] = true;                       !Flag specified now
;  229         1287  2 !
;  230         1288  2 !    To determine the maximum number of history records for new library,
;  231         1289  2 !    if /COMPRESS=HISTORY:n specified then its value will be used,
;  232         1290  2 !    else use attribute from old library header.
;  233         1291  2 !
;  234         1292  2 IF NOT .lib$gl_cre8flags [lib$c_opt_luhs]
;  235         1293  2 THEN
;  236         1294  2     lib$gl_allhis = .header [lhd$w_maxluhrec];
;  237       P 1295  2 perform (lbr$ini_control (outlibindex, func,                     !Init the control index
;  238       P 1296  2             lib$gl_type, outnamblk), lib$_initerr,
;  239         1297  2             1, outdesc);
;  240         1298  2
;  241         1299  2 ! If the user specified  /COMPRESS, (not /DATA), and the input library
;  242         1300  2 ! is already reduced, keep it that way.
;  243         1301  2 !
;  244         1302  2 IF NOT .lib$gl_ctlmsk [lib$v_data]
;  245         1303  2   AND .header [lhd$l_dcxmapvbn] neq 0
;  246         1304  2 THEN
;  247         1305  2     lib$gl_cre8flags [lib$c_opt_dcx] =1 ;
;  248         1306  2
;  249         1307  2 ! If we're creating a DCX-processed library...
;  250         1308  2 !
;  251         1309  2 IF .lib$gl_cre8flags[lib$c_opt_dcx]
;  252         1310  2 THEN
;  253         1311  2     perform( lbr$dcx_map(lib$gl_libctl, dcx_map_desc ));
;  254         1312  2
;  255         1313  2 CH$MOVE  (dsc$c_s_bln, lib$gl_libfdb [fdb$l_defext],            !Set default ext.
;  256         1314  2             lib$gl_outfdb [fdb$l_defext]);
;  257         1315  2 outnamblk [nam$l_rlf] = libnamblk;                              !Set up related filename block
;  258         1316  2 !
;  259         1317  2 !    Save size of additional data area in module if /COMP=KEEP
;  260         1318  2 !
;  261         1319  2 usrmodhdrlen = .lib$al_hdrlen [.lib$gl_type];                  !Save defaults
;  262         1320  2
;  263         1321  2 IF .lib$gl_ctlmsk [lib$v_keep]
;  264         1322  2 THEN
;  265         1323  2     lib$al_hdrlen [.lib$gl_type] = .header [lhd$b_mhdusz];     !Use value in library
;  266         1324  2
;  267         1325  2 ! Create output library; make it with data reduced if it should be so.
;  268         1326  2 !
;  269       P 1327  2 perform (lib_create_lib (.lib$gl_outfdb, outlibindex,
;  270       P 1328  2         (IF .lib$gl_cre8flags[lib$c_opt_dcx] THEN dcx_map_desc
;  271         1329  2                                             ELSE 0) ));
;  272         1330  2 lib$al_hdrlen [.lib$gl_type] = .usrmodhdrlen;                   !Restore defaults
;  273         1331  2
;  274         1332  2 IF .lib$gl_ctlmsk [lib$v_convert]                                !If this is forced convert
;  275         1333  2 THEN SIGNAL (lib$_cnvrting, 2, outdesc, libdesc);               ! tell user whats happening
;  276         1334  2 !
;  277          1335  2 ! Call the library procedures to return each entry in the module name
;  278          1336  2 ! index.  It will call copymodule for each entry.
;  279          1337  2 !
;  280        P 1338  2 rms_perform (get_index_if_not_empty(),
;  281          1339  2               lib$_indexerr, .lbr$gl_rmsstv, 1, libdesc);
;  282          1340  2 IF .lib$gl_ctlmsk [lib$v_keep]                              !If history is to be retained then
;  283          1341  2 THEN
;  284          1342  3     BEGIN
;  285          1343  3     status = lbr$get_history (lib$gl_libctl, lib_put_history);  !copy history
;  286          1344  3     IF NOT .status
;  287          1345  3     THEN
;  288          1346  3         SIGNAL (lib$_histerr, 1, libdesc, .status);
;  289          1347  2     END;
;  290          1348  2
;  291        P 1349  2 rms_perform (lbr$close (outlibindex),                      !Close the new library
;  292          1350  2               lib$_closeout, .lbr$gl_rmsstv, 1, outdesc);
;  293          1351  2
;  294        P 1352  2 rms_perform (lbr$close (lib$gl_libctl),                    !and the old library
;  295          1353  2               lib$_closein, .lbr$gl_rmsstv, 1, outdesc);
;  296          1354  2
;  297          1355  2 lib$gl_ctlmsk [lib$v_oldlib] = false;                     !No longer old library
;  298          1356  2 lib$gl_libfdb = .lib$gl_outfdb;                           !Make the library FDB
;  299          1357  2                                                           ! the old output FDB
;  300        P 1358  2 perform (lbr$ini_control (lib$gl_libctl, after_func,      !Init control block to open lib
;  301        P 1359  2               lib$gl_type, outnamblk),
;  302          1360  2               lib$_initerr, 1, outdesc);
;  303          1361  2
;  304        P 1362  2 rms_perform (lbr$open (lib$gl_libctl),                     !Open newly created library
;  305          1363  2               lib$_openin, .lbr$gl_rmsstv, 1, outdesc);
;  306          1364  2
;  307          1365  2 RETURN true
;  308          1366  1 END;                                         !Of lib_compress_lib

                                                           .TITLE  LIB_COMPRESS
                                                           .IDENT  \V04-000\

                                                           .PSECT  $OWN$,NOEXE,2

                                               00000 CURINDEX:
                                                           .BLKB   4
                                               00004 NEWTXTRFA:
                                                           .BLKB   8
                                               0000C OUTLIBINDEX:
                                                           .BLKB   4
                                    00000000   00010 FUNC:     .LONG   0                                    ;

                                                           .PSECT  $GLOBAL$,NOEXE,2

                                               00000 DCX_MAP_DESC::
                                                           .BLKB   8

                                                           .EXTRN  LIB_GET_MEM, LBR$DCX_MAP
                                                           .EXTRN  LBR$GET_HISTORY
                                                           .EXTRN  LBR$PUT_HISTORY
                                                           .EXTRN  LBR$GET_INDEX, LBR$FIND
                                                           .EXTRN  LBR$LOOKUP_KEY, LBR$INSERT_KEY
```

```
                                                        .EXTRN  LBR$PUT_END, LBR$SET_INDEX
                                                        .EXTRN  LBR$SET_MODULE, LBR$GET_RECORD
                                                        .EXTRN  LBR$PUT_RECORD, LBR$SEARCH
                                                        .EXTRN  LBR$OPEN, LBR$CLOSE
                                                        .EXTRN  LBR$INI_CONTROL
                                                        .EXTRN  LBR$INSERT_TIME
                                                        .EXTRN  LIB_LOG_OP, LIB_CREATE_LIB
                                                        .EXTRN  LBR$GL_CONTROL, LBR$GL_RMSSTV
                                                        .EXTRN  LIB$GL_TYPE, LIB$AL_HDRLEN
                                                        .EXTRN  LIB$AL_ASCBINF, LIB$GL_KEYSIZE
                                                        .EXTRN  LIB$GL_LIBCTL, LIB$GL_LIBFDB
                                                        .EXTRN  LIB$GL_OUTFDB, LIB$GL_CTLMSK
                                                        .EXTRN  LIB$GL_CRE8FLAGS
                                                        .EXTRN  LIB$GL_ALLGBLS, LIB$GL_ALLMODS
                                                        .EXTRN  LIB$GL_ALLKSZ, LIB$GL_ALLHIS
                                                        .EXTRN  LIB$GL_OBJGSDIX
                                                        .EXTRN  LIB$GL_MODNAMIX
                                                        .EXTRN  LBR$_NOLIDX, LIB$_EMPTYLIBRARY
                                                        .EXTRN  LIB$_CNVRTING, LIB$_HISTERR
                                                        .EXTRN  LIB$_INDEXERR, LIB$_INITERR
                                                        .EXTRN  LIB$_INSERTED, LIB$_INSERTERR
                                                        .EXTRN  LIB$_LOOKUPERR, LIB$_MHDERR

                                                        .PSECT  $CODE$,NOWRT,2

                                     0FFC 00000         .ENTRY  LIB_COMPRS_LIB, Save R2,R3,R4,R5,R6,R7,R8,-  ; 1221
                                                                R9,R10,R11                                    ;
                5B     0000G  CF             10  C1 00002       ADDL3   #16, LIB$GL_LIBFDB, R11                      ; 1259
                59     0000G  CF             10  C1 00008       ADDL3   #16, LIB$GL_OUTFDB, R9                       ; 1260
                5A     0000G  CF 00000040    8F  C1 0000E       ADDL3   #64, LIB$GL_LIBFDB, R10                      ; 1261
                58     0000G  CF 00000040    8F  C1 00018       ADDL3   #64, LIB$GL_OUTFDB, R8                       ; 1262
                              50 00000000G   00  D0 00022       MOVL    LBR$GL_CONTROL, R0                           ; 1268
                              56          0A A0  D0 00029       MOVL    10(R0), HEADER                               ;
                08     0000G  CF             01  E0 0002D       BBS     #1, LIB$GL_CRE8FLAGS, 1$                     ; 1269
  0000G         CF        6A  A6         6E  A6  C3 00033       SUBL3   110(HEADER), 106(HEADER), LIB$GL_ALLGBLS     ; 1270
                08     0000G  CF             02  E0 0003B 1$:   BBS     #2, LIB$GL_CRE8FLAGS, 2$                     ; 1271
  0000G         CF        6E  A6         78  A6  C1 00041       ADDL3   120(HEADER), 110(HEADER), LIB$GL_ALLMODS     ; 1272
                24     0000G  CF             03  E0 00049 2$:   BBS     #3, LIB$GL_CRE8FLAGS, 4$                     ; 1273
                                  0000G  CF  95 0004F           TSTB    LIB$GL_CTLMSK+2                              ; 1274
                                         0F  18 00053           BGEQ    3$                                           ;
                              50  0000G  CF  D0 00055           MOVL    LIB$GL_TYPE, R0                              ; 1275
                       0000G  CF  0000GCF40  D0 0005A           MOVL    LIB$AL_ASCBINF[R0], LIB$GL_ALLKSZ            ;
                                         0F  11 00062           BRB     4$                                           ;
                              50   00C4  C6  9E 00064 3$:       MOVAB   196(HEADER), R0                              ; 1281
                       0000G  CF     02  A0  3C 00069           MOVZWL  2(R0), LIB$GL_ALLKSZ                         ; 1283
                                  0000G  CF  D7 0006F           DECL    LIB$GL_ALLKSZ                                ;
                       0000G  CF  0000G  CF  D0 00073 4$:       MOVL    LIB$GL_ALLKSZ, LIB$GL_KEYSIZE                ; 1285
                       0000G  CF         08  88 0007A           BISB2   #8, LIB$GL_CRE8FLAGS                         ; 1286
                06     0000G  CF         04  E0 0007F           BBS     #4, LIB$GL_CRE8FLAGS, 5$                     ; 1292
                       0000G  CF     7C  A6  3C 00085           MOVZWL  124(HEADER), LIB$GL_ALLHIS                   ; 1294
                                         58  DD 0008B 5$:       PUSHL   R8                                           ; 1297
                                  0000G  CF  9F 0008D           PUSHAB  LIB$GL_TYPE                                  ;
                                  0000'  CF  9F 00091           PUSHAB  FUNC                                         ;
                                  0000'  CF  9F 00095           PUSHAB  OUTLIBINDEX                                  ;
                   00000000G  00         04  FB 00099           CALLS   #4, LBR$INI_CONTROL                          ;
                              13         50  E8 000A0           BLBS    STATUS, 6$                                   ;
                                         50  DD 000A3           PUSHL   STATUS                                       ;
                                  59  DD 000A5          PUSHL   R9                                   ;
                                  01  DD 000A7          PUSHL   #1                                   ;
                       00000000G  8F  DD 000A9          PUSHL   #LIB$_INITERR                        ;
          00000000G  00           04  FB 000AF          CALLS   #4, LIB$SIGNAL                       ;
   0C         0000G  CF           03  E0 000B6 6$:      BBS     #3, LIB$GL_CTLMSK+4, 7$              ; 1302
                           008C   C6  D5 000BC          TSTL    140(HEADER)                          ; 1303
                                  06  13 000C0          BEQL    7$                                   ;
              0000G  CF      80   8F  88 000C2          BISB2   #128, LIB$GL_CRE8FLAGS               ; 1305
                           0000G  CF  95 000C8 7$:      TSTB    LIB$GL_CRE8FLAGS                     ; 1309
                                  12  18 000CC          BGEQ    8$                                   ;
                           0000'  CF  9F 000CE          PUSHAB  DCX_MAP_DESC                         ; 1311
                           0000G  CF  9F 000D2          PUSHAB  LIB$GL_LIBCTL                        ;
          00000000G  00           02  FB 000D6          CALLS   #2, LBR$DCX_MAP                      ;
                     48           50  E9 000DD          BLBC    STATUS, 12$                          ;
                     50    0000G  CF  D0 000E0 8$:      MOVL    LIB$GL_LIBFDB, R0                    ; 1313
                     57    0000G  CF  D0 000E5          MOVL    LIB$GL_OUTFDB, R7                    ; 1314
   08    A7          08    A0     08  28 000EA          MOVC3   #8, 8(R0), 8(R7)                     ;
                     10    A8     5A  D0 000F0          MOVL    R10, 16(R8)                          ; 1315
                     50    0000G  CF  D0 000F4          MOVL    LIB$GL_TYPE, R0                      ; 1319
                     52    0000GCF40  D0 000F9          MOVL    LIB$AL_HDRLEN[R0], USRMODHDRLEN      ;
                           0000G  CF  95 000FF          TSTB    LIB$GL_CTLMSK+3                      ; 1321
                                  07  18 00103          BGEQ    9$                                   ;
              0000GCF40      3C   A6  9A 00105          MOVZBL  60(HEADER), LIB$AL_HDRLEN[R0]        ; 1323
                           0000G  CF  95 0010C 9$:      TSTB    LIB$GL_CRE8FLAGS                     ; 1329
                                  09  18 00110          BGEQ    10$                                  ;
                     50    0000'  CF  9E 00112          MOVAB   DCX_MAP_DESC, R0                     ;
                                  50  DD 00117          PUSHL   R0                                   ;
                                  02  11 00119          BRB     11$                                  ;
                                  7E  D4 0011B 10$:     CLRL    -(SP)                                ;
                           0000'  CF  9F 0011D 11$:     PUSHAB  OUTLIBINDEX                          ;
                                  57  DD 00121          PUSHL   R7                                   ;
              0000G  CF           03  FB 00123          CALLS   #3, LIB_CREATE_LIB                   ;
                     01           50  E8 00128 12$:     BLBS    STATUS, 13$                          ;
                                      04 0012B          RET                                          ;
                     50    0000G  CF  D0 0012C 13$:     MOVL    LIB$GL_TYPE, R0                      ; 1330
              0000GCF40           52  D0 00131          MOVL    USRMODHDRLEN, LIB$AL_HDRLEN[R0]      ;
                     13    0000G  CF  E9 00137          BLBC    LIB$GL_CTLMSK+3, 14$                 ; 1332
                           0A00   8F  BB 0013C          PUSHR   #^M<R9,R11>                          ; 1333
                                  02  DD 00140          PUSHL   #2                                   ;
                       00000000G  8F  DD 00142          PUSHL   #LIB$_CNVRTING                       ;
          00000000G  00           04  FB 00148          CALLS   #4, LIB$SIGNAL                       ;
              0000V  CF           00  FB 0014F 14$:     CALLS   #0, GET_INDEX_IF_NOT_EMPTY           ; 1339
                     19           50  E8 00154          BLBS    STATUS, 15$                          ;
                       00000000G  00  DD 00157          PUSHL   LBR$GL_RMSSTV                        ;
                                  50  DD 0015D          PUSHL   STATUS                               ;
                                  5B  DD 0015F          PUSHL   R11                                  ;
                                  01  DD 00161          PUSHL   #1                                   ;
                       00000000G  8F  DD 00163          PUSHL   #LIB$_INDEXERR                       ;
          00000000G  00           05  FB 00169          CALLS   #5, LIB$SIGNAL                       ;
                           0000G  CF  95 00170 15$:     TSTB    LIB$GL_CTLMSK+3                      ; 1340
                                  25  18 00174          BGEQ    16$                                  ;
                           0000V  CF  9F 00176          PUSHAB  LIB_PUT_HISTORY                      ; 1343
                           0000G  CF  9F 0017A          PUSHAB  LIB$GL_LIBCTL                        ;
          00000000G  00           02  FB 0017E          CALLS   #2, LBR$GET_HISTORY                  ;
                     13           50  E8 00185          BLBS    STATUS, 16$                          ; 1344
                                  50  DD 00188          PUSHL   STATUS                               ; 1346
                                  5B  DD 0018A          PUSHL   R11                                  ;
```

```
                                01  DD 0018C          PUSHL   #1                          ;
                     00000000G  8F  DD 0018E          PUSHL   #LIB$_HISTERR               ;
         00000000G  00          04  FB 00194          CALLS   #4, LIB$SIGNAL              ;
                         0000'  CF  9F 0019B  16$:    PUSHAB  OUTLIBINDEX                 ; 1350
         00000000G  00          01  FB 0019F          CALLS   #1, LBR$CLOSE               ;
                    19          50  E8 001A6          BLBS    STATUS, 17$                 ;
                     00000000G  00  DD 001A9          PUSHL   LBR$GL_RMSSTV               ;
                                50  DD 001AF          PUSHL   STATUS                      ;
                                59  DD 001B1          PUSHL   R9                          ;
                                01  DD 001B3          PUSHL   #1                          ;
                      00861058  8F  DD 001B5          PUSHL   #8786008                    ;
         00000000G  00          05  FB 001BB          CALLS   #5, LIB$SIGNAL              ;
                         0000G  CF  9F 001C2  17$:    PUSHAB  LIB$GL_LIBCTL               ; 1353
         00000000G  00          01  FB 001C6          CALLS   #1, LBR$CLOSE               ;
                    19          50  E8 001CD          BLBS    STATUS, 18$                 ;
                     00000000G  00  DD 001D0          PUSHL   LBR$GL_RMSSTV               ;
                                50  DD 001D6          PUSHL   STATUS                      ;
                                59  DD 001D8          PUSHL   R9                          ;
                                01  DD 001DA          PUSHL   #1                          ;
                      00861050  8F  DD 001DC          PUSHL   #8786000                    ;
         00000000G  00          05  FB 001E2          CALLS   #5, LIB$SIGNAL              ;
             0000G  CF      80  8F  8A 001E9  18$:    BICB2   #128, LIB$GL_CTLMSK+2       ; 1355
             0000G  CF   0000G  CF  D0 001EF          MOVL    LIB$GL_OUTFDB, LIB$GL_LIBFDB ; 1356
                                58  DD 001F6          PUSHL   R8                          ; 1360
                         0000G  CF  9F 001F8          PUSHAB  LIB$GL_TYPE                 ;
                            04  AC  9F 001FC          PUSHAB  AFTER_FUNC                  ;
                         0000G  CF  9F 001FF          PUSHAB  LIB$GL_LIBCTL               ;
         00000000G  00          04  FB 00203          CALLS   #4, LBR$INI_CONTROL         ;
                    13          50  E8 0020A          BLBS    STATUS, 19$                 ;
                                50  DD 0020D          PUSHL   STATUS                      ;
                                59  DD 0020F          PUSHL   R9                          ;
                                01  DD 00211          PUSHL   #1                          ;
                     00000000G  8F  DD 00213          PUSHL   #LIB$_INITERR               ;
         00000000G  00          04  FB 00219          CALLS   #4, LIB$SIGNAL              ;
                         0000G  CF  9F 00220  19$:    PUSHAB  LIB$GL_LIBCTL               ; 1363
         00000000G  00          01  FB 00224          CALLS   #1, LBR$OPEN                ;
                    19          50  E8 0022B          BLBS    STATUS, 20$                 ;
                     00000000G  00  DD 0022E          PUSHL   LBR$GL_RMSSTV               ;
                                50  DD 00234          PUSHL   STATUS                      ;
                                59  DD 00236          PUSHL   R9                          ;
                                01  DD 00238          PUSHL   #1                          ;
                      00861098  8F  DD 0023A          PUSHL   #8786072                    ;
         00000000G  00          05  FB 00240          CALLS   #5, LIB$SIGNAL              ;
                    50          01  D0 00247  20$:    MOVL    #1, R0                      ; 1365
                                    04 0024A          RET                                 ; 1366

; Routine Size:  587 bytes,    Routine Base:  $CODE$ + 0000
;  310        1367  1 ROUTINE get_index_if_not_empty =
;  311        1368  2 BEGIN
;  312        1369  2 LOCAL
;  313        1370  2     status;
;  314        1371  2 !
;  315        1372  2 ! Call the library procedures to return each entry in the module name
;  316        1373  2 ! index.  It will call copymodule for each entry.  Treat the empty library
;  317        1374  2 ! case benignly.
;  318        1375  2 !
;  319        1376  2 status = lbr$get_index (lib$gl_libctl, lib$gl_modnamix,          !Return the index
;  320        1377  2                 copymodule);                                    !and call copymodule for each entry
;  321        1378  2
;  322        1379  2 IF .status EQL lbr$_nulidx THEN
;  323        1380  3     BEGIN
;  324        1381  3     signal (lib$_emptylibrary, 1, lib$gl_libfdb[fdb$l_namdesc]);
;  325        1382  3     status = ss$_normal;
;  326        1383  2     END;
;  327        1384  2
;  328        1385  2 RETURN .status;
;  329        1386  1 END;



                                                   0004 00000 GET_INDEX_IF_NOT_EMPTY:
                                                                          .WORD   Save R2                                  ; 1367
                                         0000V  CF   9F 00002             PUSHAB  COPYMODULE                               ; 1376
                                         0000G  CF   9F 00006             PUSHAB  LIB$GL_MODNAMIX                          ;
                                         0000G  CF   9F 0000A             PUSHAB  LIB$GL_LIBCTL                            ;
                         00000000G  00          03   FB 0000E             CALLS   #3, LBR$GET_INDEX                        ;
                                    52          50   D0 00015             MOVL    R0, STATUS                               ;
                         00000000G  8F          52   D1 00018             CMPL    STATUS, #LBR$_NULIDX                     ; 1379
                                                18   12 0001F             BNEQ    1$                                       ;
                     7E       0000G  CF         10   C1 00021             ADDL3   #16, LIB$GL_LIBFDB, -(SP)                ; 1381
                                                01   DD 00027             PUSHL   #1                                       ;
                                   00000000G    8F   DD 00029             PUSHL   #LIB$_EMPTYLIBRARY                       ;
                         00000000G  00          03   FB 0002F             CALLS   #3, LIB$SIGNAL                           ;
                                    52          01   D0 00036             MOVL    #1, STATUS                               ; 1382
                                    50          52   D0 00039 1$:         MOVL    STATUS, R0                               ; 1385
                                                     04 0003C             RET                                              ; 1386

; Routine Size:  61 bytes,    Routine Base:  $CODE$ + 024B
```

```
  1 ROUTINE copymodule (keydesc, modrfa) =
  2 BEGIN
  2
  2 !++
  2 ! This routine is called by the librarian for each name in the
  2 ! module name index.  The text for the name is read and inserted
  2 ! into the new library, and then the key is inserted into the
  2 ! index.  If there is more than one index in the library, the other
  2 ! indices are searched to find all symbols associated with
  2 ! the module, and they are entered into the new library.
  2 !
  2 ! Inputs:
  2 !
  2 !       keydesc         address of string descriptor for module name
  2 !       modrfa          address of rfa of module
  2 !
  2 ! Outputs:
  2 !
  2 !       The module is copied into the output library
  2 !
  2 !--
  2
  2 MAP
  2     keydesc : REF BBLOCK [dsc$c_s_bln];
  2
  2 LOCAL
  2     rms_status,                                     !Status from RMS operations
  2     first_put,                                      !flag true when first put done
  2     header : BBLOCK [lbr$c_pagesize],               !Buffer for header
  2     bufdesc : BBLOCK [dsc$c_s_bln],                 !descriptor for buffer
  2     rfa : BBLOCK [rfa$c_length];                    !Dummy RFA
  2
  2 BIND
  2     libheader = .lbr$gl_control [lbr$l_hdrptr] : BBLOCK,!Point to the library header
  2     libdesc = lib$gl_libfdb [fdb$l_namdesc] : BBLOCK,   !Name the filename descriptor
  2     outdesc = lib$gl_outfdb [fdb$l_namdesc] : BBLOCK;   !...
  2
P 2 rms_perform (lbr$find (lib$gl_libctl, .modrfa),         !Lookup key to find text
  2             lib$_lookuperr, .lbr$gl_rmsstv, 2, .keydesc, libdesc);
  2
  2 bufdesc [dsc$a_pointer] = header;
  2 first_put = true;
  2
  2 !
  2 ! Read all text records for the module until end of file is returned.  Write the records
  2 ! into the new library.
  2 !
  3 WHILE (bufdesc [dsc$w_length] = lbr$c_pagesize;
  3         rms_status = lbr$get_record (lib$gl_libctl, bufdesc, bufdesc); !Read all records of module
  4         IF NOT .rms_status AND (.rms_status NEQ rms$_eof)
  4         THEN BEGIN
  4             SIGNAL (lib$_readerr, 1, libdesc, .rms_status, .lbr$gl_rmsstv);
  4             EXITLOOP;
  3             END;
  3
  3         .rms_status NEQ rms$_eof)
  3 DO BEGIN
```

```
;     3         LOCAL
;     3             status;
;     3         status = lbr$put_record (outlibindex, bufdesc,              ! and write them to new library
;     3                         (IF .first_put THEN newtxtrfa ELSE rfa));
;     3         IF NOT .status
;     4         THEN BEGIN                                  ! exit and write end of module record
;     4             signal(lib$_writeerr, 1, outdesc, .status, .lbr$gl_rmsstv);
;     4             EXITLOOP;
;     3             END;
;     3
;     3     first_put = false;
;     2     END;
;     2 !
;     2 ! Text for module has been copied. Write end of module record
;     2 !
;   P 2 rms_perform (lbr$put_end (outlibindex),          !Terminate PUT
;     2             lib$_writeerr, .lbr$gl_rmsstv, 1, outdesc);
;     2 !
;     2 ! Insert the module name into the new library
;     2 !
;   P 2 perform (lbr$set_index (outlibindex, lib$gl_modnamix),  !Insert into module name index
;     2             lib$_indexerr, 1, outdesc);
;     2
;   P 2 rms_perform (lbr$insert_key (outlibindex, .keydesc, newtxtrfa), !Insert key into index
;     2             lib$_inserterr, .lbr$gl_rmsstv, 2, .keydesc, outdesc);
;     2
;     2 !
;     2 ! Read module header from old library
;     2 !
;     2 bufdesc [dsc$w_length] = lbr$c_maxhdrsiz;
;     2 bufdesc [dsc$a_pointer] = header;
;   P 2 rms_perform (lbr$set_module (lib$gl_libctl, .modrfa, bufdesc, bufdesc),
;     2         lib$_mhderr, .lbr$gl_rmsstv, 2, .keydesc, libdesc);
;     2 !
;     2 ! Set insert date/time of module in new library
;     2 !
;     2 lbr$insert_time (outlibindex, newtxtrfa, header [mhd$l_datim]);
;   P 2 perform (lbr$set_index (lib$gl_libctl, lib$gl_modnamix),        !Set to old library
;     2             lib$_indexerr, 1, libdesc);
;     2 !
;     2 ! If there is user information in the module header, update the module header
;     2 ! in the new library.
;     2 !
;     2 IF .libheader [lhd$b_mhdusz] NEQ 0
;     3 THEN BEGIN
;     3     bufdesc [dsc$w_length] = .libheader  [lhd$b_mhdusz];                   !Set length of update data
;     3     bufdesc [dsc$a_pointer] = header [mhd$b_usrdat];                       !Point to update data
;   P 3     rms_perform (lbr$set_module (outlibindex, newtxtrfa,0,0,bufdesc),   !Update module header
;     3             lib$_mhderr, .lbr$gl_rmsstv, 2, .keydesc, outdesc);
;   P 3     perform  (lbr$set_index (lib$gl_libctl, lib$gl_modnamix),            !Set to old library
;     3             lib$_indexerr, 1, libdesc);
;     2     END;
;     2 !
;     2 ! If there are global symbols in the module, then search the index of the old library for them
;     2 ! so they can be entered into the new library global symbol index
;     2 !
;     2 IF .libheader [lhd$b_nindex] GTR 1                       !If there is more than one index
;   445        1501  3 THEN BEGIN
;   446        1502  3     INCRU i FROM 2 TO .libheader [lhd$b_nindex]          !Loop through the other indices
;   447        1503  4     DO BEGIN
;   448        1504  4         curindex = .i;                               !Set current index number
;   449      P 1505  4         rms_perform (lbr$search (lib$gl_libctl,      !Search index for symbols
;   450      P 1506  4                 curindex, .modrfa, enterglobals),    !so they can be entered in new library
;   451        1507  4                 lib$_indexerr, .lbr$gl_rmsstv, 1, libdesc);
;   452        1508  4
;   453        1509  3         END;
;   454        1510  2     END;
;   455        1511  2
;   456      P 1512  2 perform (lbr$set_index (lib$gl_libctl, lib$gl_modnamix),!Do set index to set index number and lbr$gl_control
;   457        1513  2                 lib$_indexerr, 1, libdesc);
;   458        1514  2
;   459        1515  2 lib_log_op (lib$_inserted, .keydesc, .lib$gl_outfdb);   !Log on console if logging
;   460        1516  2
;   461        1517  2 RETURN true
;   462        1518  1 END;                                         !Of copymodule



                                             0FFC 00000 COPYMODULE:
                                                                   .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11         ; 1387
                  5B 00000000G   00  9E 00002                      MOVAB    LBR$SET_INDEX, R11                           ;
                  5A 00000000G   8F  D0 00009                      MOVL     #LIB$_INDEXERR, R10                          ;
                  59     0000'   CF  9E 00010                      MOVAB    OUTLIBINDEX, R9                              ;
                  58 00000000G   00  9E 00015                      MOVAB    LBR$GL_RMSSTV, R8                            ;
                  57 00000000G   00  9E 0001C                      MOVAB    LIB$SIGNAL, R7                               ;
                  5E      FDF0   CE  9E 00023                      MOVAB    -528(SP), SP                                 ;
                  50 00000000G   00  D0 00028                      MOVL     LBR$GL_CONTROL, R0                           ; 1420
                  52        0A   A0  D0 0002F                      MOVL     10(R0), R2                                   ;
 55       0000G   CF             10  C1 00033                      ADDL3    #16, LIB$GL_LIBFDB, R5                       ; 1421
 56       0000G   CF             10  C1 00039                      ADDL3    #16, LIB$GL_OUTFDB, R6                       ; 1422
                            08   AC  DD 0003F                      PUSHL    MODRFA                                       ; 1425
                         0000G   CF  9F 00042                      PUSHAB   LIB$GL_LIBCTL                                ;
      00000000G   00             02  FB 00046                      CALLS    #2, LBR$FIND                                 ;
                  14             50  E8 0004D                      BLBS     STATUS, 1$                                   ;
                                 68  DD 00050                      PUSHL    LBR$GL_RMSSTV                                ;
                                 50  DD 00052                      PUSHL    STATUS                                       ;
                                 55  DD 00054                      PUSHL    R5                                           ;
                            04   AC  DD 00056                      PUSHL    KEYDESC                                      ;
                                 02  DD 00059                      PUSHL    #2                                           ;
                     00000000G   8F  DD 0005B                      PUSHL    #LIB$_LOOKUPERR                              ;
                  67             06  FB 00061                      CALLS    #6, LIB$SIGNAL                               ;
         0C       AE        10   AE  9E 00064 1$:                  MOVAB    HEADER, BUFDESC+4                            ; 1427
                  54             01  D0 00069                      MOVL     #1, FIRST_PUT                                ; 1428
         08       AE      0200   8F  B0 0006C 2$:                  MOVW     #512, BUFDESC                                ; 1434
                            08   AE  9F 00072                      PUSHAB   BUFDESC                                      ; 1435
                            0C   AE  9F 00075                      PUSHAB   BUFDESC                                      ;
                         0000G   CF  9F 00078                      PUSHAB   LIB$GL_LIBCTL                                ;
      00000000G   00             03  FB 0007C                      CALLS    #3, LBR$GET_RECORD                           ;
                  53             50  D0 00083                      MOVL     R0, RMS_STATUS                               ;
                  19             53  E8 00086                      BLBS     RMS_STATUS, 3$                               ; 1436
      0001827A    8F             53  D1 00089                      CMPL     RMS_STATUS, #98938                           ;
                                 10  13 00090                      BEQL     3$                                           ;
```

```
                               68 DD 00092         PUSHL   LBR$GL_RMSSTV              ; 1438
                               53 DD 00094         PUSHL   RMS_STATUS                 ;
                               55 DD 00096         PUSHL   R5                         ;
                               01 DD 00098         PUSHL   #1                         ;
                    008610B2   8F DD 0009A         PUSHL   #8786098                   ;
                               34 11 000A0         BRB     6$                         ;
 0001827A    8F                53 D1 000A2 3$:     CMPL    RMS_STATUS, #98938         ; 1442
                               34 13 000A9         BEQL    8$                         ;
             06                54 E9 000AB         BLBC    FIRST_PUT, 4$              ; 1447
             50          F8    A9 9E 000AE         MOVAB   NEWTXTRFA, R0              ;
                               03 11 000B2         BRB     5$                         ;
             50                6E 9E 000B4 4$:     MOVAB   RFA, R0                    ;
                               50 DD 000B7 5$:     PUSHL   R0                         ;
                         0C    AE 9F 000B9         PUSHAB  BUFDESC                    ; 1446
                               59 DD 000BC         PUSHL   R9                         ;
00000000G    00                03 FB 000BE         CALLS   #3, LBR$PUT_RECORD         ;
             13                50 E8 000C5         BLBS    STATUS, 7$                 ; 1448
                               68 DD 000C8         PUSHL   LBR$GL_RMSSTV              ; 1450
                               50 DD 000CA         PUSHL   STATUS                     ;
                               56 DD 000CC         PUSHL   R6                         ;
                               01 DD 000CE         PUSHL   #1                         ;
                    008610D2   8F DD 000D0         PUSHL   #8786130                   ;
             67                05 FB 000D6 6$:     CALLS   #5, LIB$SIGNAL             ;
                               04 11 000D9         BRB     8$                         ; 1449
                               54 D4 000DB 7$:     CLRL    FIRST_PUT                  ; 1454
                               8D 11 000DD         BRB     2$                         ; 1434
                               59 DD 000DF 8$:     PUSHL   R9                         ; 1460
00000000G    00                01 FB 000E1         CALLS   #1, LBR$PUT_END            ;
             11                50 E8 000E8         BLBS    STATUS, 9$                 ;
                               68 DD 000EB         PUSHL   LBR$GL_RMSSTV              ;
                               50 DD 000ED         PUSHL   STATUS                     ;
                               56 DD 000EF         PUSHL   R6                         ;
                               01 DD 000F1         PUSHL   #1                         ;
                    008610D2   8F DD 000F3         PUSHL   #8786130                   ;
             67                05 FB 000F9         CALLS   #5, LIB$SIGNAL             ;
                       0000G   CF 9F 000FC 9$:     PUSHAB  LIB$GL_MODNAMIX            ; 1465
                               59 DD 00100         PUSHL   R9                         ;
             6B                02 FB 00102         CALLS   #2, LBR$SET_INDEX          ;
             0B                50 E8 00105         BLBS    STATUS, 10$                ;
                               50 DD 00108         PUSHL   STATUS                     ;
                               56 DD 0010A         PUSHL   R6                         ;
                               01 DD 0010C         PUSHL   #1                         ;
                               5A DD 0010E         PUSHL   R10                        ;
             67                04 FB 00110         CALLS   #4, LIB$SIGNAL             ;
                         F8    A9 9F 00113 10$:    PUSHAB  NEWTXTRFA                  ; 1468
             53          04    AC D0 00116         MOVL    KEYDESC, R3                ;
                               53 DD 0011A         PUSHL   R3                         ;
                               59 DD 0011C         PUSHL   R9                         ;
00000000G    00                03 FB 0011E         CALLS   #3, LBR$INSERT_KEY         ;
             13                50 E8 00125         BLBS    STATUS, 11$                ;
                               68 DD 00128         PUSHL   LBR$GL_RMSSTV              ;
                               50 DD 0012A         PUSHL   STATUS                     ;
                       0048    8F BB 0012C         PUSHR   #^M<R3,R6>                 ;
                               02 DD 00130         PUSHL   #2                         ;
                   00000000G   8F DD 00132         PUSHL   #LIB$_INSERTERR            ;
             67                06 FB 00138         CALLS   #6, LIB$SIGNAL             ;
       08    AE          80    8F 9B 0013B 11$:    MOVZBW  #128, BUFDESC              ; 1473
       0C  AE         10  AE  9E 00140         MOVAB   HEADER, BUFDESC+4              ; 1474
                      08  AE  9F 00145         PUSHAB  BUFDESC                        ; 1476
                      0C  AE  9F 00148         PUSHAB  BUFDESC                        ;
                      08  AC  DD 0014B         PUSHL   MODRFA                         ;
                   0000G  CF  9F 0014E         PUSHAB  LIB$GL_LIBCTL                  ;
00000000G  00             04  FB 00152         CALLS   #4, LBR$SET_MODULE             ;
           11             50  E8 00159         BLBS    STATUS, 12$                    ;
                          68  DD 0015C         PUSHL   LBR$GL_RMSSTV                  ;
                          50  DD 0015E         PUSHL   STATUS                         ;
                          28  BB 00160         PUSHR   #^M<R3,R5>                     ;
                          02  DD 00162         PUSHL   #2                             ;
               00000000G  8F  DD 00164         PUSHL   #LIB$_MHDERR                   ;
           67             06  FB 0016A         CALLS   #6, LIB$SIGNAL                 ;
                      18  AE  9F 0016D  12$:   PUSHAB  HEADER+8                       ; 1480
                      F8  A9  9F 00170         PUSHAB  NEWTXTRFA                      ;
                          59  DD 00173         PUSHL   R9                             ;
00000000G  00             03  FB 00175         CALLS   #3, LBR$INSERT_TIME            ;
                   0000G  CF  9F 0017C         PUSHAB  LIB$GL_MODNAMIX                ; 1482
                   0000G  CF  9F 00180         PUSHAB  LIB$GL_LIBCTL                  ;
           6B             02  FB 00184         CALLS   #2, LBR$SET_INDEX              ;
           0B             50  E8 00187         BLBS    STATUS, 13$                    ;
                          50  DD 0018A         PUSHL   STATUS                         ;
                          55  DD 0018C         PUSHL   R5                             ;
                          01  DD 0018E         PUSHL   #1                             ;
                          5A  DD 00190         PUSHL   R10                            ;
           67             04  FB 00192         CALLS   #4, LIB$SIGNAL                 ;
                      3C  A2  95 00195  13$:   TSTB    60(R2)                         ; 1487
                          4A  13 00198         BEQL    15$                            ;
       08  AE         3C  A2  9B 0019A         MOVZBW  60(R2), BUFDESC                ; 1489
       0C  AE         20  AE  9E 0019F         MOVAB   HEADER+16, BUFDESC+4           ; 1490
                      08  AE  9F 001A4         PUSHAB  BUFDESC                        ; 1492
                          7E  7C 001A7         CLRQ    -(SP)                          ;
                      F8  A9  9F 001A9         PUSHAB  NEWTXTRFA                      ;
                          59  DD 001AC         PUSHL   R9                             ;
00000000G  00             05  FB 001AE         CALLS   #5, LBR$SET_MODULE             ;
           13             50  E8 001B5         BLBS    STATUS, 14$                    ;
                          68  DD 001B8         PUSHL   LBR$GL_RMSSTV                  ;
                          50  DD 001BA         PUSHL   STATUS                         ;
                    0048  8F  BB 001BC         PUSHR   #^M<R3,R6>                     ;
                          02  DD 001C0         PUSHL   #2                             ;
               00000000G  8F  DD 001C2         PUSHL   #LIB$_MHDERR                   ;
           67             06  FB 001C8         CALLS   #6, LIB$SIGNAL                 ;
                   0000G  CF  9F 001CB  14$:   PUSHAB  LIB$GL_MODNAMIX                ; 1494
                   0000G  CF  9F 001CF         PUSHAB  LIB$GL_LIBCTL                  ;
           6B             02  FB 001D3         CALLS   #2, LBR$SET_INDEX              ;
           0B             50  E8 001D6         BLBS    STATUS, 15$                    ;
                          50  DD 001D9         PUSHL   STATUS                         ;
                          55  DD 001DB         PUSHL   R5                             ;
                          01  DD 001DD         PUSHL   #1                             ;
                          5A  DD 001DF         PUSHL   R10                            ;
           67             04  FB 001E1         CALLS   #4, LIB$SIGNAL                 ;
           52         01  A2  9A 001E4  15$:   MOVZBL  1(R2), R2                      ; 1500
           01             52  91 001E8         CMPB    R2, #1                         ;
                          35  1B 001EB         BLEQU   19$                            ;
           54             02  D0 001ED         MOVL    #2, I                          ; 1502
                          2B  11 001F0         BRB     18$                            ;
       F4  A9             54  D0 001F2  16$:   MOVL    I, CURINDEX                    ; 1504
```

```
                            0000V   CF  9F 001F6          PUSHAB  ENTERGLOBALS                         ; 1507
                               08   AC  DD 001FA          PUSHL   MODRFA                               ;
                               F4   A9  9F 001FD          PUSHAB  CURINDEX                             ;
                            0000G   CF  9F 00200          PUSHAB  LIB$GL_LIBCTL                        ;
               00000000G  00        04  FB 00204          CALLS   #4, LBR$SEARCH                       ;
                          0D        50  E8 0020B          BLBS    STATUS, 17$                          ;
                                    68  DD 0020E          PUSHL   LBR$GL_RMSSTV                        ;
                                    50  DD 00210          PUSHL   STATUS                               ;
                                    55  DD 00212          PUSHL   R5                                   ;
                                    01  DD 00214          PUSHL   #1                                   ;
                                    5A  DD 00216          PUSHL   R10                                  ;
                          67        05  FB 00218          CALLS   #5, LIB$SIGNAL                       ;
                                    54  D6 0021B   17$:   INCL    I                                    ; 1502
                          52        54  D1 0021D   18$:   CMPL    I, R2                                ;
                                    D0  1B 00220          BLEQU   16$                                  ;
                            0000G   CF  9F 00222   19$:   PUSHAB  LIB$GL_MODNAMIX                      ; 1513
                            0000G   CF  9F 00226          PUSHAB  LIB$GL_LIBCTL                        ;
                          6B        02  FB 0022A          CALLS   #2, LBR$SET_INDEX                    ;
                          0B        50  E8 0022D          BLBS    STATUS, 20$                          ;
                                    50  DD 00230          PUSHL   STATUS                               ;
                                    55  DD 00232          PUSHL   R5                                   ;
                                    01  DD 00234          PUSHL   #1                                   ;
                                    5A  DD 00236          PUSHL   R10                                  ;
                          67        04  FB 00238          CALLS   #4, LIB$SIGNAL                       ;
                            0000G   CF  DD 0023B   20$:   PUSHL   LIB$GL_OUTFDB                        ; 1515
                                    53  DD 0023F          PUSHL   R3                                   ;
                        00000000G   8F  DD 00241          PUSHL   #LIB$_INSERTED                       ;
                   0000G  CF        03  FB 00247          CALLS   #3, LIB_LOG_OP                       ;
                          50        01  D0 0024C          MOVL    #1, R0                               ; 1517
                                        04 0024F          RET                                          ; 1518

; Routine Size:  592 bytes,    Routine Base:  $CODE$ + 0288


;  463          1519  1
;  464          1520  1 ROUTINE lib_put_history (rec_desc) =
;  465          1521  2 BEGIN
;  466          1522  2 !++
;  467          1523  2 !
;  468          1524  2 !--
;  469          1525  2 RETURN lbr$put_history ( outlibindex, .rec_desc );
;  470          1526  1 END;     ! of lib_put_history




                                  0000 00000 LIB_PUT_HISTORY:
                                                          .WORD   Save nothing                         ; 1520
                               04   AC  DD 00002          PUSHL   REC_DESC                             ; 1525
                            0000'   CF  9F 00005          PUSHAB  OUTLIBINDEX                          ;
               00000000G  00        02  FB 00009          CALLS   #2, LBR$PUT_HISTORY                  ;
                                        04 00010          RET                                          ; 1526

; Routine Size:  17 bytes,    Routine Base:  $CODE$ + 04D8
;   472          1527  1 ROUTINE enterglobals (keydesc) =
;   473          1528  2 BEGIN
;   474          1529  2 !++
;   475          1530  2 !
;   476          1531  2 ! This routine is called to enter a global symbol into the global symbol
;   477          1532  2 ! index for an object module
;   478          1533  2 !
;   479          1534  2 ! Inputs:
;   480          1535  2 !
;   481          1536  2 !      keydesc         address of descriptor for symbol name
;   482          1537  2 !
;   483          1538  2 ! Outputs:
;   484          1539  2 !
;   485          1540  2 !      Global symbol name is entered into index of new library
;   486          1541  2 !
;   487          1542  2 !--
;   488          1543  2
;   489          1544  2 MAP
;   490          1545  2     keydesc : REF BBLOCK;                      !Really a string descriptor
;   491          1546  2 BIND
;   492          1547  2     libdesc = lib$gl_libfdb [fdb$l_namdesc] : BBLOCK,   !Name the filename descriptor
;   493          1548  2     outdesc = lib$gl_outfdb [fdb$l_namdesc] : BBLOCK;   !...
;   494          1549  2
;   495        P 1550  2 perform (lbr$set_index (outlibindex, curindex),
;   496          1551  2             lib$_indexerr, 1, outdesc);
;   497          1552  2
;   498        P 1553  2 rms_perform (lbr$insert_key (outlibindex, .keydesc, newtxtrfa),
;   499          1554  2             lib$_inserterr, .lbr$gl_rmsstv, 2, .keydesc, outdesc);
;   500          1555  2
;   501        P 1556  2 perform (lbr$set_index (lib$gl_libctl, lib$gl_modnamix),
;   502          1557  2             lib$_indexerr, 1, libdesc);
;   503          1558  2
;   504          1559  2 RETURN true
;   505          1560  1 END;                                     !of enterglobals





                                     00FC 00000 ENTERGLOBALS:
                                                    .WORD    Save R2,R3,R4,R5,R6,R7                 ; 1527
                  57 00000000G  8F   D0 00002       MOVL     #LIB$_INDEXERR, R7                     ;
                  56 00000000G  00   9E 00009       MOVAB    LBR$SET_INDEX, R6                      ;
                  55     0000'  CF   9E 00010       MOVAB    OUTLIBINDEX, R5                        ;
                  54 00000000G  00   9E 00015       MOVAB    LIB$SIGNAL, R4                         ;
       53      0000G  CF        10   C1 0001C       ADDL3    #16, LIB$GL_LIBFDB, R3                 ; 1547
       52      0000G  CF        10   C1 00022       ADDL3    #16, LIB$GL_OUTFDB, R2                 ; 1548
                           F4   A5   9F 00028       PUSHAB   CURINDEX                               ; 1551
                                55   DD 0002B       PUSHL    R5                                     ;
                  66            02   FB 0002D       CALLS    #2, LBR$SET_INDEX                      ;
                  0B            50   E8 00030       BLBS     STATUS, 1$                             ;
                                50   DD 00033       PUSHL    STATUS                                 ;
                                52   DD 00035       PUSHL    R2                                     ;
                                01   DD 00037       PUSHL    #1                                     ;
                                57   DD 00039       PUSHL    R7                                     ;
                  64            04   FB 0003B       CALLS    #4, LIB$SIGNAL                         ;
                           F8   A5   9F 0003E 1$:   PUSHAB   NEWTXTRFA                              ; 1554
```

```
                          04    AC  DD 00041           PUSHL   KEYDESC                          ;
                                55  DD 00044           PUSHL   R5                               ;
              00000000G   00    03  FB 00046           CALLS   #3, LBR$INSERT_KEY               ;
                          18    50  E8 0004D           BLBS    STATUS, 2$                       ;
                  00000000G     00  DD 00050           PUSHL   LBR$GL_RMSSTV                    ;
                                50  DD 00056           PUSHL   STATUS                           ;
                                52  DD 00058           PUSHL   R2                               ;
                          04    AC  DD 0005A           PUSHL   KEYDESC                          ;
                                02  DD 0005D           PUSHL   #2                               ;
                  00000000G     8F  DD 0005F           PUSHL   #LIB$_INSERTERR                  ;
                          64    06  FB 00065           CALLS   #6, LIB$SIGNAL                   ;
                      0000G     CF  9F 00068 2$:       PUSHAB  LIB$GL_MODNAMIX                  ; 1557
                      0000G     CF  9F 0006C           PUSHAB  LIB$GL_LIBCTL                    ;
                          66    02  FB 00070           CALLS   #2, LBR$SET_INDEX                ;
                          0B    50  E8 00073           BLBS    STATUS, 3$                       ;
                                50  DD 00076           PUSHL   STATUS                           ;
                                53  DD 00078           PUSHL   R3                               ;
                                01  DD 0007A           PUSHL   #1                               ;
                                57  DD 0007C           PUSHL   R7                               ;
                          64    04  FB 0007E           CALLS   #4, LIB$SIGNAL                   ;
                          50    01  D0 00081 3$:       MOVL    #1, R0                           ; 1559
                                    04 00084           RET                                      ; 1560

; Routine Size:  133 bytes,    Routine Base:  $CODE$ + 04E9


;  506          1561  1 END                             ! Of module
;  507          1562  0 ELUDOM



                                                    .EXTRN  LIB$SIGNAL

;
;                              PSECT SUMMARY
;
;       Name                    Bytes                       Attributes
;
;   $GLOBAL$                        8  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $OWN$                          20  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                       1390  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;
;                         Library Statistics
;
;                                           -------- Symbols --------       Pages        Processing
;       File                                Total    Loaded   Percent       Mapped       Time
;
;   _$255$DUA28:[SYSLIB]STARLET.L32;1        9776        32         0        581          00:01.1
;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:COMPRESS/OBJ=OBJ$:COMPRESS MSRC$:COMPRESS/UPDATE=(ENH$:COMPRESS)

; Size:          1390 code + 28 data bytes
; Run Time:         00:30.1
; Elapsed Time:     00:59.8
; Lines/CPU Min:      3117
; Lexemes/CPU-Min: 34449
; Memory Used:  262 pages
; Compilation Complete
```

0200 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY